

A scope text editing program

N. E. Wiseman

One of the most obviously useful devices for two-way communication with a computer is an electric typewriter or teleprinter and many conversational mode experiments have been implemented on systems employing typewriter consoles. Another device, just emerging from experimental status, is a cathode ray tube display coupled with some position-indicating mechanism such as a light pen or tracker ball. The computer can display and animate arbitrarily complex pictures on the CRT and the user can input data using the light pen. There are many applications, involving for instance descriptions of shapes or connected networks or mechanisms in motion, which utilize pictorial data of the sort which can readily be passed across an interface of this sort. If we consider some activity in natural language processing like say abstracting or editing or group listing, we find many instances of operations which are strongly sensitive to context. To carry out this activity using a computer in conversational mode will frequently require this context information to pass across the man/machine interface. In order for the computer to request help with the interpretation of a single character, a hundred or more characters of context may have to be shown to the man, and this 'picture' of the context will contain somewhere (not necessarily, or even usually, at the end) the character under discussion. To converse with the computer about such a character using a typewriter is obviously a very tiresome and unnatural thing. A CRT display on the other hand provides a fast and convenient method. 4,000 or more characters can be displayed on the screen at electronic speeds and a character or phrase or line can be seen and 'discussed' in context using markers or pointers put up by the program.

The text editor program is intended primarily for use on the sort of text found in a computing laboratory (programs, data and typewritten reports). In its present form, it can edit in, and transliterate between, any of the character codes we use at Cambridge, including a full upper/lower case Flexowriter code. By making minor changes to the character subroutines and program steering tables any new code could be accommodated.

A paper tape copy of the document to be edited is passed through the program which maintains a circular buffer of about 6,000 characters. If addition, or more input, to the text is made when the buffer is full, the head of the buffer is automatically punched out to make room for the tail. In this way arbitrarily large documents may be edited although access at any moment is limited to the most recent 6,000 characters. Any part of the buffer may be selected and displayed on the CRT display which then serves as a 'window' into the text. Editing is always done in the window. The user can read the context, select the editing point, modify the text and then see what he has done. The edit point is

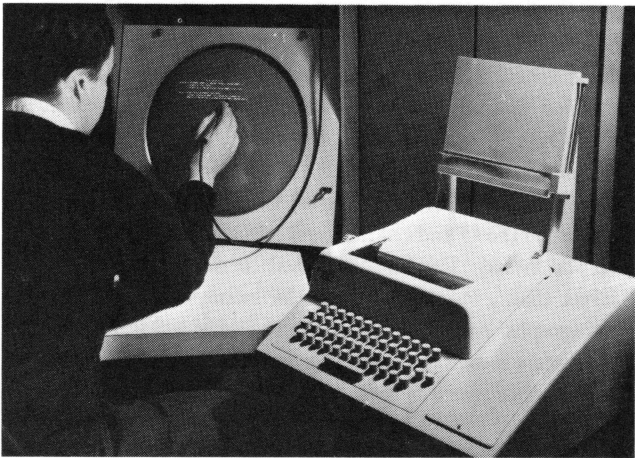


Fig 1 PDP7 and 338 display and light pen with input keyboard

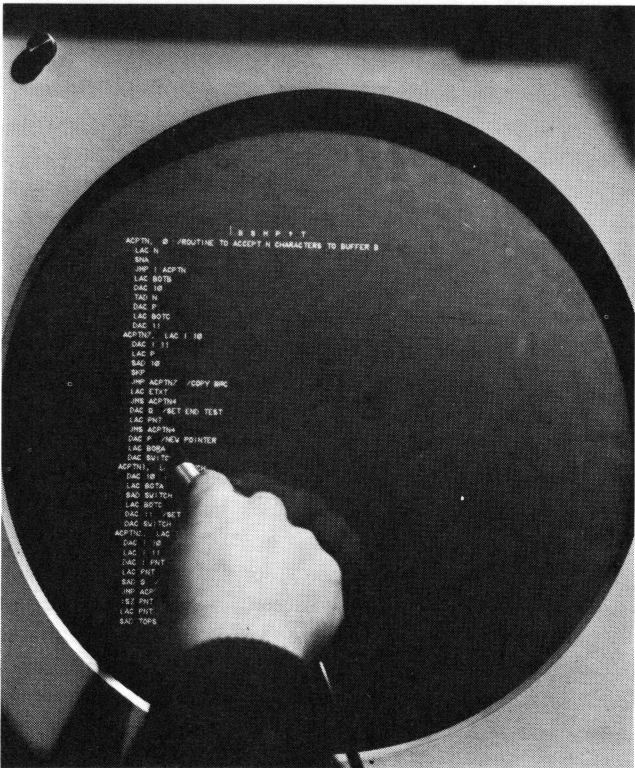


Fig 2 338 display and light pen

indicated by an underline symbol which can be moved anywhere in the window by keyboard commands or by pointing with the light pen. Other commands serve to move the window, punch the buffer, print on the teleprinter, search the text or carry out some kind of edit function.

Near the top of the screen of the CRT, above the area used as text window, eight mode identifying characters I B S M P ↑ T C are displayed. These are light buttons. Pointing with the light pen at one of these characters will cause it to enlarge to double size and the program to switch into the corresponding mode. Alternatively the mode can be selected by typing ALTMODE followed by the mode identifier. I shall now outline the actions in each of the eight modes.

I (character insert): Creation of new text and insertions, deletions and changes over a small area of existing text would normally be carried out in I-mode. Characters typed on the teleprinter are inserted into the text to the right of the current pointer position and the pointer is then stepped on. The character RUBOUT, however, deletes the character above the pointer and steps the pointer back.

For example to change COP to CLAP the pointer is set to underline the O and characters RUBOUT, L, A are typed.

B (block insert): This mode is used to manipulate text items larger than a single character—for example phrases, lines or paragraphs. An invisible buffer capable of holding up to 256 characters is used to store a block of text and typed commands serve to process the main text using this buffer. The commands are:

K Kill invisible buffer and reset counts M, N.

E Extract text from the right of the pointer position up to the next line feed into the invisible buffer, advance the pointer and count in M and N.

I Insert N characters from the invisible buffer into the main text to the right of the pointer position and advance the pointer.

RUBOUT: Delete M characters in the main text and back up the pointer.

D Move what lies between the pointer and the end of the current line on to the beginning of the next line, hyphenating any word broken by the pointer.

U Move what lies between the beginning of the current line and the pointer onto the end of the line above, dehyphenating as necessary.

: Search for line starting <string>. See under S-mode.

* Search for <string>. See under S-mode.

S (string search). A string of characters may be typed on the teleprinter and a forwards search through the text from the pointer made for characters which match the string. The search is initiated by typing ':' or '*', ':' looking for a line starting <string> and '*' looking for <string> anywhere. If a match is found the window is re-positioned so that the line containing the match is at the top and the pointer is set under the last character in the string. Typing ':' or '*' again moves on to the next occurrence. While typing the string the M counter is incremented. RUBOUT however moves the window back to the beginning of the buffer and sets M to zero.

M (move window): Typing U or D causes the text to move up or down by 20 lines (ie about half a screen diameter), and repositions the pointer to underline the first character in the window. In this mode the program endeavours to maintain a window full of text by reading in paper tape as necessary. If a form feed is encountered in reading (indicating a page ending) the program switches to mode S and stops the reader.

P (punch text): The contents of the text buffer up to and including the pointer position are punched out and terminated by a form feed.

↑ (print text): This is used to print parts of the text on the teleprinter. Printing starts at the pointer and continues until the end of the buffer, or the pointer is again reached (having been repositioned to 'bracket' the text desired) or the mode is changed.

T (type to buffer): Characters typed on the teleprinter are added into the invisible buffer and counted in N. Rubout deletes the last character and decreases the count by 1.

C (copy): Form feed is used by convention to indicate end of block (or page, or chapter). On entering C-mode a number of blocks from 1 to 9 are copied from the source document to the output without disturbing the text buffers. The number of blocks is assumed to be 1 unless a number (2-9) is typed before the first block is completed.

As an example of the use of the editor, consider the replacing of occurrences of BIG DOG by HORSE: type BIG DOG in mode S, HORSE in mode T and enter mode B. Type '*' to find an occurrence of BIG DOG (leaves text as BIG DOG), type RUBOUT to delete BIG DOG and 'I' to insert HORSE. '*', RUBOUT, 'I', '*', RUBOUT, 'I' etc. will find and change successive occurrences in the same way.

α (ALT-MODE): This is not itself a mode but the editor goes into a limbo state in which certain extra non-editing operations are permitted. Typing α at any time causes the editor to enter limbo, or leave it if it is already in. Any command undefined in limbo state causes escape to the normal mode. The defined commands are: I B S M P ↑ T C. These have already been mentioned. They cause escape from limbo into the mode corresponding to the particular mode identifier.

L R D U: These commands move the pointer left, right, down, up, respectively, but do not cause escape.

RUBOUT: This positions the pointer at the end of the text.

SPACE: Punches a short length of leader tape.

Thus to move the pointer to the 3rd character on the 2nd line down from where it is and then to enter mode B, say, type D D R R R B.

The facilities provided by the editor, while adequate for the work of our laboratory, are extremely limited. For editing more general text, justifying and hyphenating routines would be added and mode-B operations would be extended to include functions sensitive to paragraph, page and chapter delimiters. Additional buffers would be provided to accommodate folios and page numbers. A change of philosophy to permit a certain amount of automatic editing would also seem desirable and an evolving system for semi-automatic page makeup could then be set up.

Acknowledgement

I should like to thank Mr K. Paton and Mr E. Thomas for the help they have given me with implementing the editor.